CIVIL INFRASTRUCTURE PLATFORM

Delta Updates
Making Updates Leaner

Felix Moessbauer, Jan Kiszka – Siemens AG,

Embedded OSS Summit Seattle, 2024

About Us

Felix Moessbauer

<felix.moessbauer@siemens.com>

- Siemens Technology
- (In-house) Embedded Linux consultant & developer
- Contributes to major OSS projects for Siemens
- Tooling developer (static / dyn. analysis, build tools, ...)

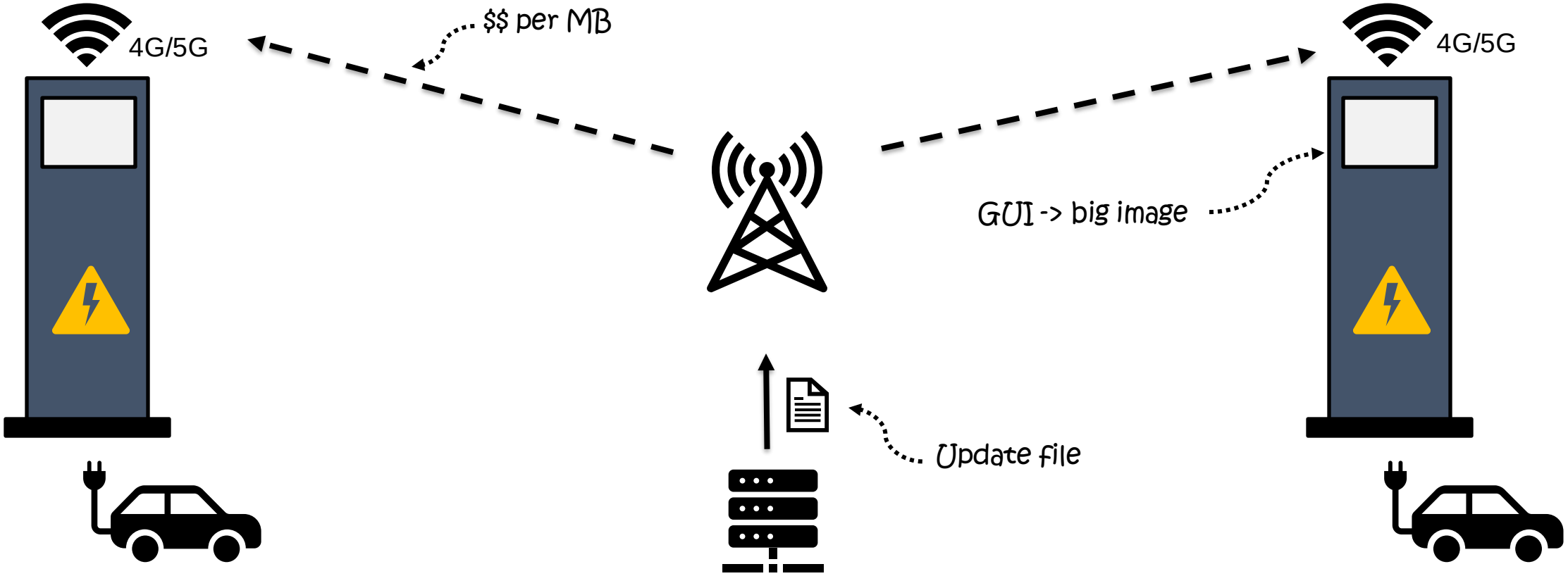
Jan Kiszka

<jan.kiszka@siemens.com>

- Siemens Technology
- (In-house) Embedded Linux consultant & developer
- CIP kernel workgroup chair, isar-cip-core maintainer
- Maintainer and contributor to various OSS projects

Robust Updating of Industrial Devices in the Field

An example use-case



Civil Infrastructure Platform (CIP) Project

Provide an industrial-grade Linux

- Build on top of existing OSS projects
- Contribute to them to fill gaps
- Create new OSS projects were needed
- Provide integration patterns
- Enhance life-time of OSS where needed (LTS)
- Prepare security certifications

CIP-Core and Software Update Workgroups

- Implement pre-integration in isar-cip-core[1]
  - Isar[2]: Debian image builder
  - Use Debian binaries where possible, build own ones where needed
- Define securely bootable & updateable reference images
- Concepts (not isar-cip-core recipes) can be transferred to Yocto/OE as well

[1] https://gitlab.com/cip-project/cip-core/isar-cip-core [2] https://github.com/ilbers/isar

CIVIL INFRASTRUCTURE PLATFORM

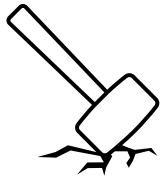
Features and Concepts in CIP Software Update

SWUpdate[1] + wfx[2]

- Software update for embedded system on device.
- Support Suricatta mode and wfx (workflow executor) for remote update.

Reliable

- Dual-copy update pattern based on Round-Robin Handler.
- EFI Boot Guard as bootloader for robust boot path selection.
- Immutable rootfs + overlay for /etc

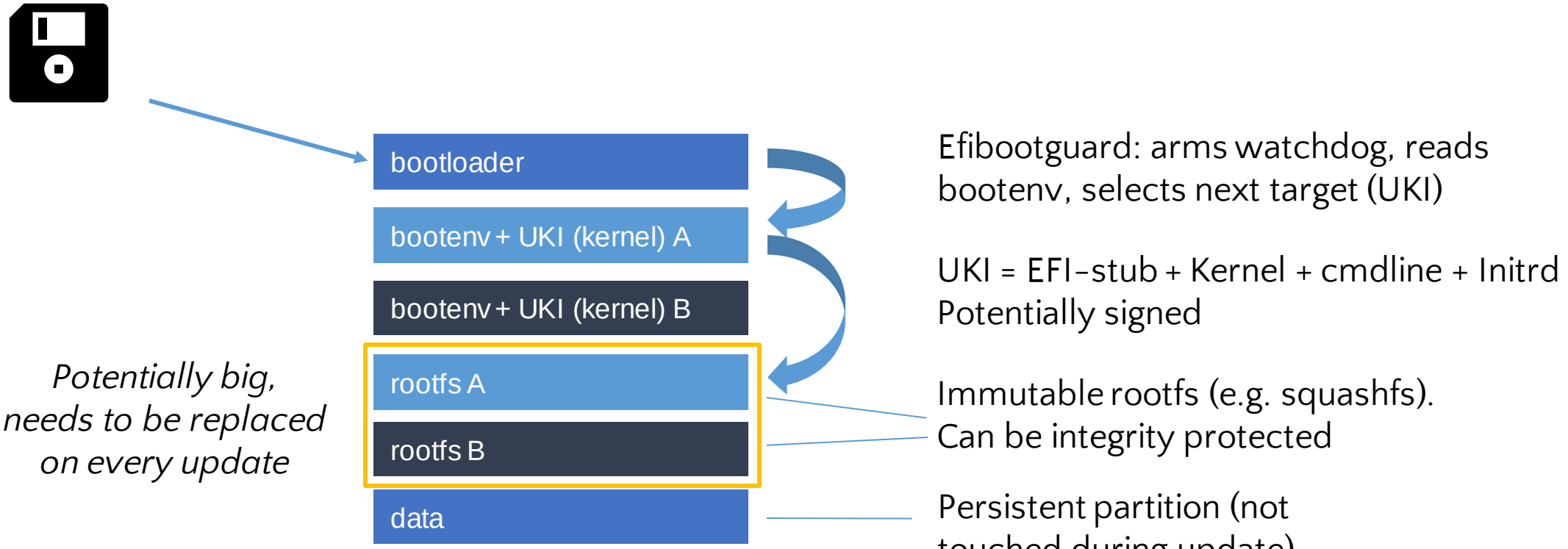


Secure

- Secure boot with EFI Boot Guard.
- SWUpdate image signing for image source verification.
- Disk encryption for data partitions (e.g. /var), optionally also root partition.
- Integrity verification of rootfs based on dm-verity.

[1] https://github.com/sbabic/swupdate [2] https://github.com/siemens/wfx

CIP A/B rootfs Update – Partition Layout



Potentially big, needs to be replaced on every update

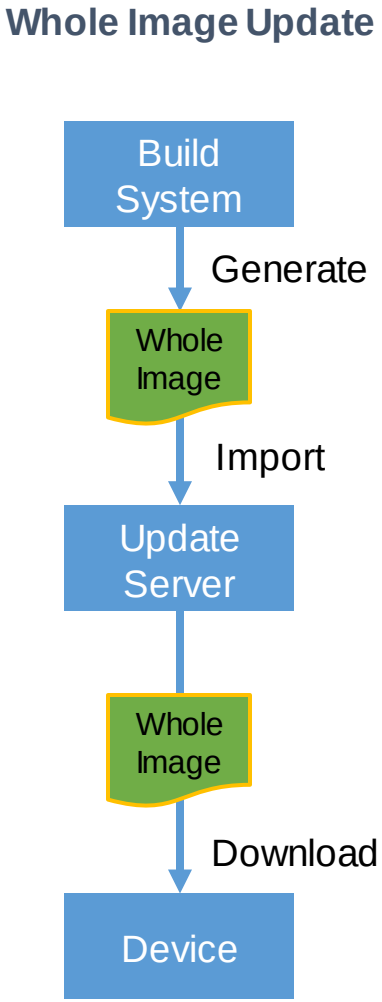
Efibootguard: arms watchdog, reads bootenv, selects next target (UKI)

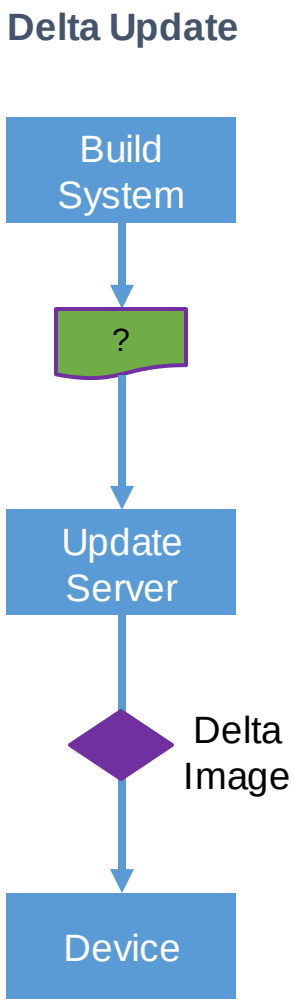
UKI = EFI-stub + Kernel + cmdline + Initrd
Potentially signed

Immutable rootfs (e.g. squashfs).
Can be integrity protected

Persistent partition (not touched during update)

Delta Update and the Differences





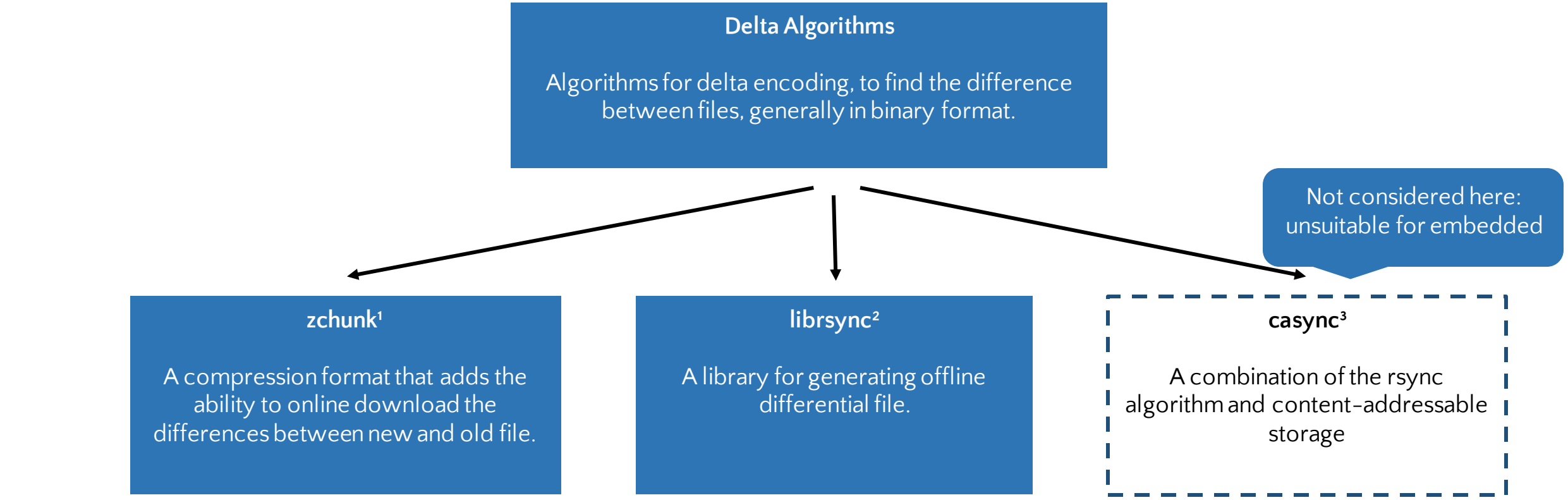
Delta update

- Only changes of the image (delta image) need to be transferred.
- Resource efficient when the whole image is big, and changes are small.
- Necessary when bandwidth for update is limited.

Questions

- Is it possible to support all the features for software update reliability and security on delta update pattern?
- What needs to be changed in the image building and update process?

Delta Algorithms: librsync, zchunk, casync



[1] https://github.com/zchunk/zchunk, [2] https://github.com/librsync/librsync, [3] https://github.com/systemd/casync
In depth comparison by Toshiba: https://docs.google.com/presentation/d/16iMgqzKczvWTufkWF_EwzaUMNhsQAmoi/

rdiff handler for SWUpdate

CIVIL INFRASTRUCTURE PLATFORM

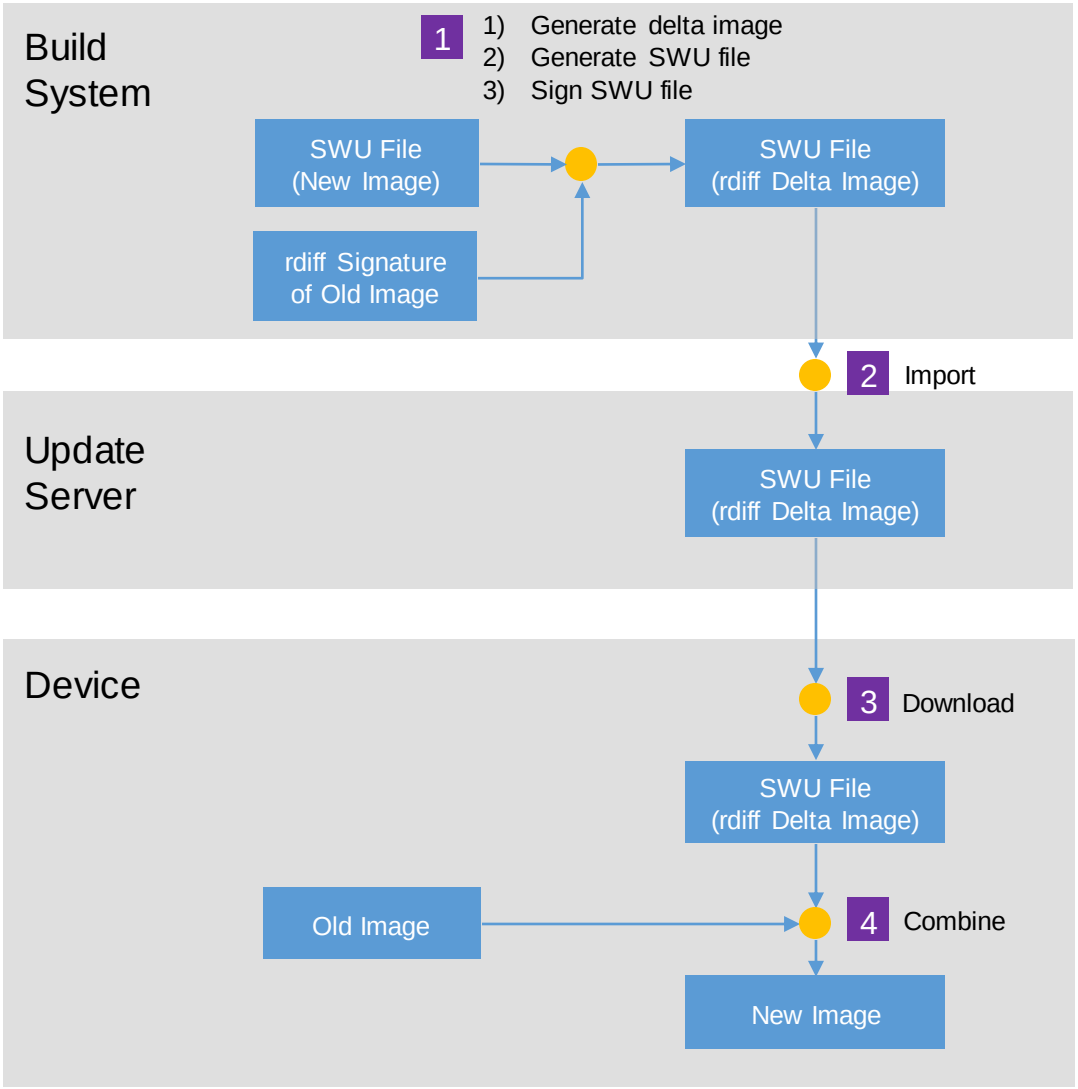
Swupdate + delta update
https://sbabic.github.io/swupdate/delta-update.html

Delta Update based on librsync / rdiff – pre-generated

Step 1: Build SWU file for delta image

Step 2: Import to update server

Step 3: Download image and update



Description

- Generate rdiff delta image in build system, from which the rdiff signature of old images should be accessible.
- Only support limited versions of delta update.

Pros

- No change needed in update server
- SWU file signing is done in build system which avoids exposing key to update server
- SWU file download process already supported by SWUpdate.
- Offline update possible (e.g., via USB drive)

Cons

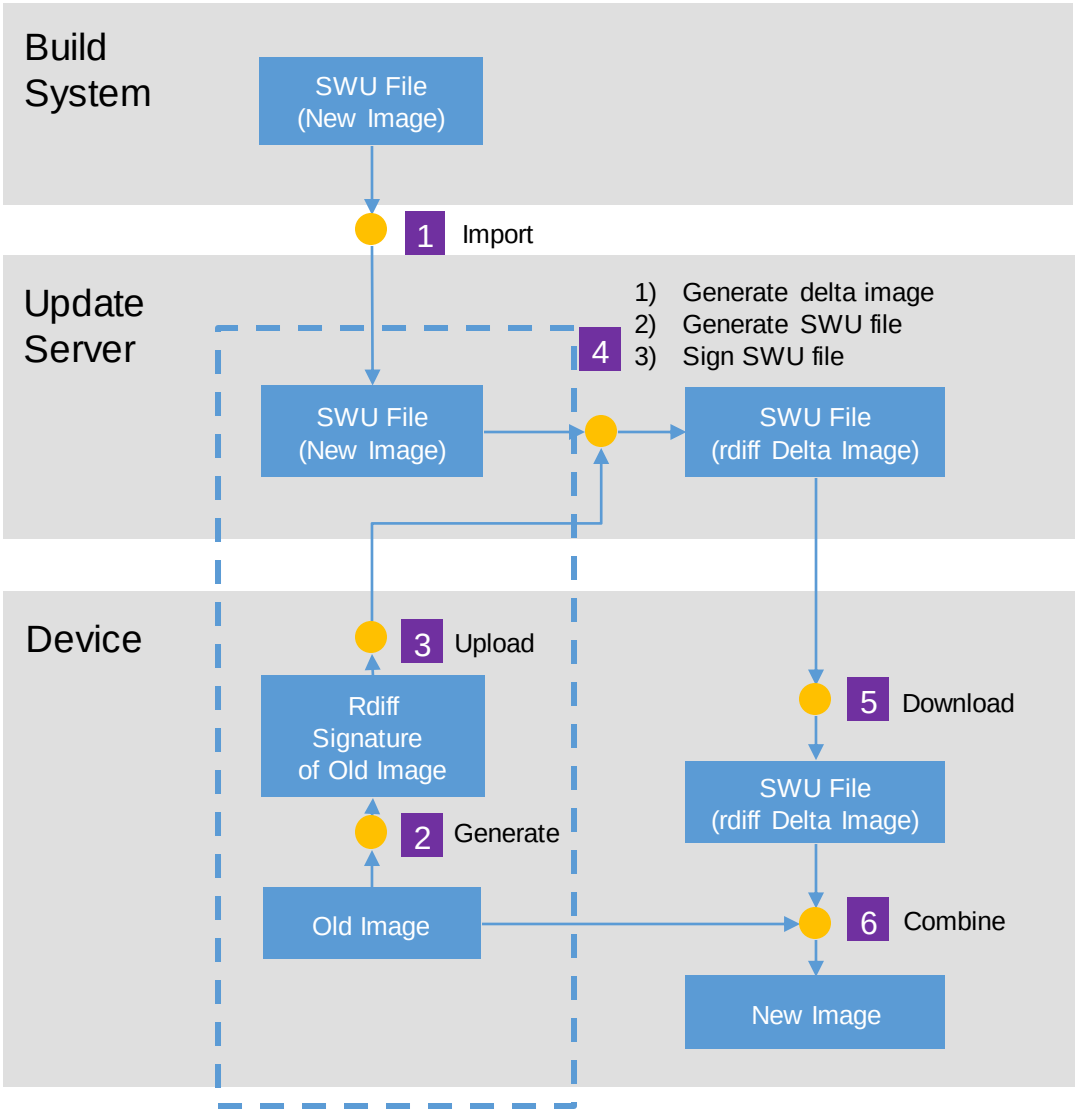
- Only incremental updates are supported (e.g. 1.0 -> 1.1 -> 1.2, but not 1.0 -> 1.2)

Delta Update using librsync / rdiff – on-demand generated

Step 1: Build SWU file for new image

Step 2: Create SWU file for delta image

Step 3: Download image and update



via API, not swupdate

Description

- Generate rdiff delta image in update server, based on rdiff signature of old image from device.

Pros

- Flexibly support different versions of images from devices.
- No change needed in build system.

Cons

- Need to upload rdiff signature file from device, which is not part of original SWUpdate workflow.
- SWU file for delta image need to be signed in update server, which might cause security issue due to exposure of certificate/key for image signing.

Zchunk handler for SWUpdate

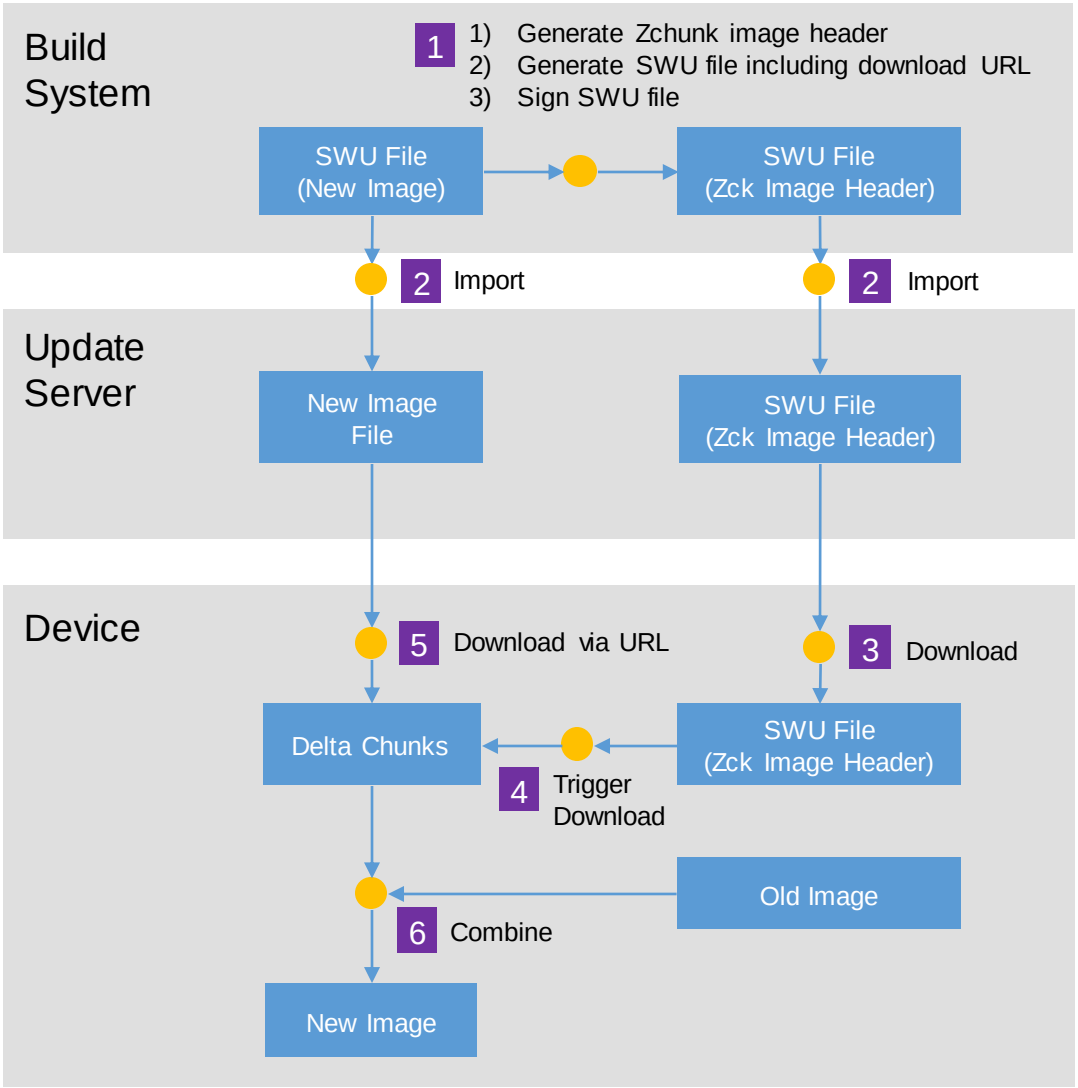
CIVIL INFRASTRUCTURE PLATFORM

Delta Update based on Zchunk

Step 1: Build SWU file for new image and delta image

Step 2: Import to update server

Step 3: Download image and update



Preconditions

- Generate Zchunk delta image in build system.
- Provide web service for downloading Zchunk delta chunks.

Pros

- Flexibly support different versions of images from devices.
- SWU file signing can be in build system which can avoid security issue in update server.
- SWU file download process already supported by SWUpdate.

Cons

- Need to import two image files and provide an additional web service for downloading delta chunks from update server, which is different from whole image update.
- Need to provide two update artifacts

Optimizing the image building

Pre-conditions to make the update small

= changed = unchanged

packages

build reproducibly

Reproducible Builds

rootfs (chroot)

Install reproducibly supported in isar-cip-core

CIVIL INFRASTRUCTURE PLATFORM

rootfs: squashfs (compressed)

compress continuously (use rsyncable algorithms)

Update artifact

metadata

Don't compress or use rsyncable algorithms. Use reasonable chunk size

Features and Concepts in CIP Software Update

isar-cip-core

- Required components ✓
- Generation of rdiff and zchunk artifacts ✓
- Released? ×

SWUpdate + Handler

- SWUpdate ✓
- Suricatta LUA Handler ✓
- Handler chaining for Round-Robin Handler with rdiff Handler & Delta Update ✓

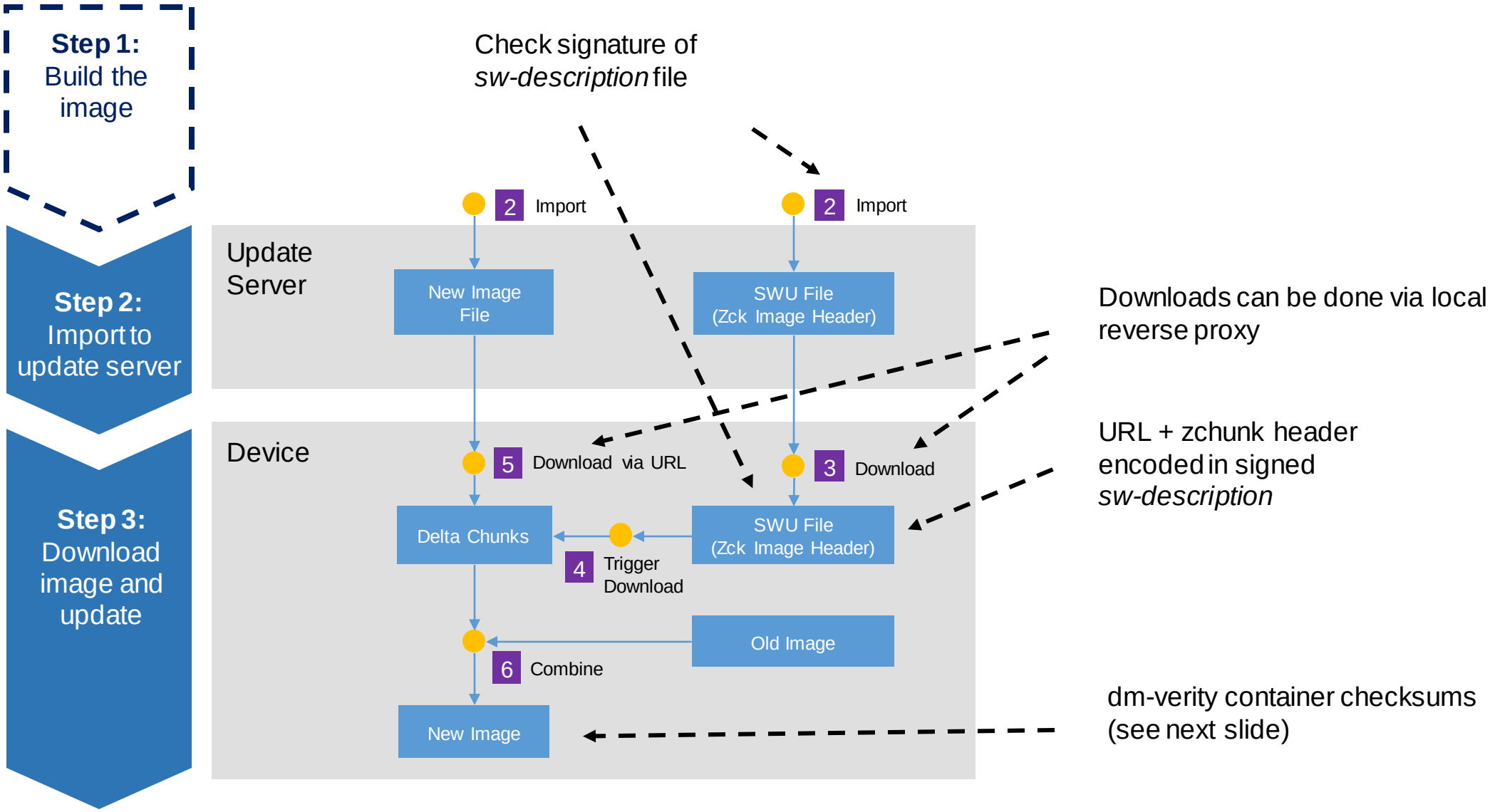
Backend Support

- rdiff: No backend change required ✓
- Zchunk
  -> Hawkbit (×)
  -> WFX ✓

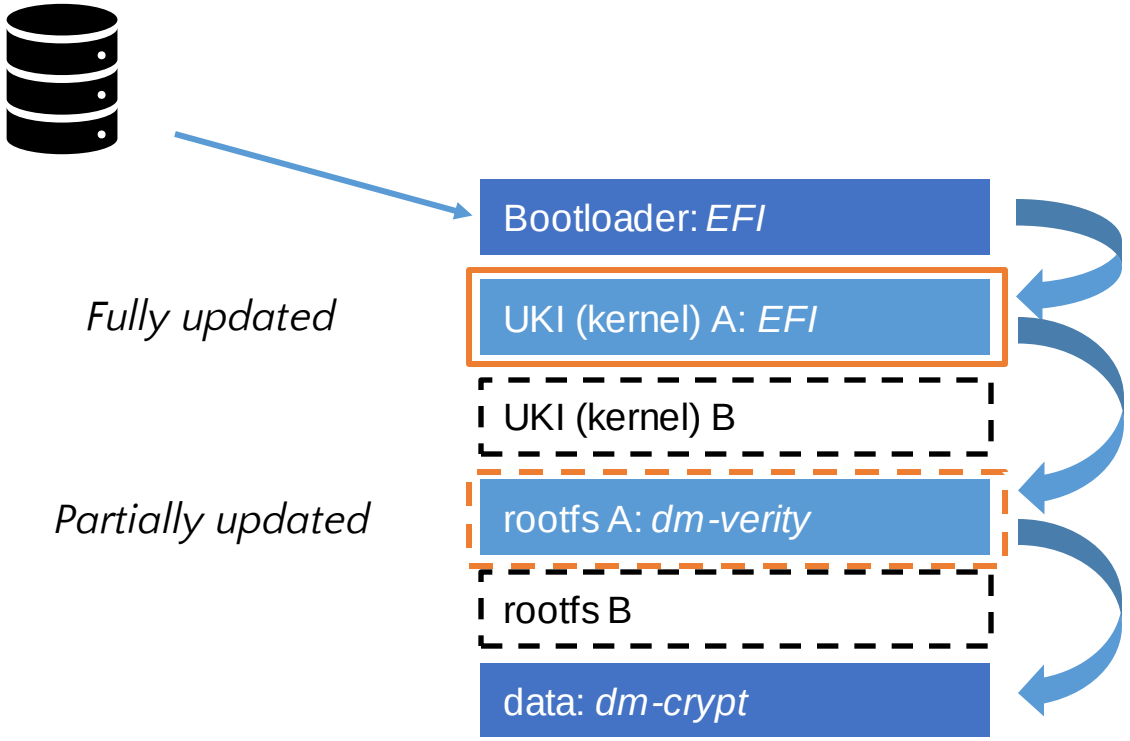
Delta Updates & Security

CIVIL INFRASTRUCTURE PLATFORM

FAQ: Securing the Update Path (zchunk)



FAQ: Delta Updates & Secure Boot – Why does it work?



Efibootguard
signed with SB keys

UKI = EFI-stub + Kernel + cmdline + **Initrd**
signed with SB keys

dm-verity **hash** encoded in initrd

dm-crypt key in hardware or
firmware TPM

Summary & Kudos

More resource efficient when the whole image is big, and changes are small

Reliable: Supports dual-copy update pattern

No impact on security: secure boot, disk encryption, and dm-verity based integrity verification.

No free lunch: Image generation and update workflows are different between rdiff and Zchunk, users need to decide which one to use according to requirements and use cases.

Siemens AG[1] / Siemens Ltd. China[2]
Felix Moessbauer[1]
Jan Kiszka[1]
Wang Qi[2]

Toshiba
Adithya Balakumar
Dinesh Kumar
Kazuhiro Hayashi

CIVIL INFRASTRUCTURE PLATFORM